REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 01-06-2007	**2. REPORT TYPE** Final Report	**3. DATES COVERED** *(From – To)* 7 February 2006 - 22-Sep-08

4. TITLE AND SUBTITLE Rapid Physics-Based Prediction of Air Vehicle Stability and Response with Application to Wing Rock	**5a. CONTRACT NUMBER** FA8655-06-1-3033
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Professor Kenneth J Badcock	**5d. PROJECT NUMBER**
	5d. TASK NUMBER
	5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Liverpool Brownlow Hill Liverpool L69 3GH United Kingdom	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) EOARD Unit 4515 BOX 14 APO AE 09421	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) Grant 06-3033

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The objective of this proposal is to develop efficient CFD based techniques for the prediction of wing rock amplitudes for a delta wing-body. These methods, based on either a centre manifold projection or the method of matched scales, use information obtained from the direct calculation of instability onset to produce a small order system of equations which can predict the limit cycle response for values of the bifurcation parameter near the onset value. The successful demonstration of these techniques for large order systems and in a robust fashion would be unique.

15. SUBJECT TERMS

EOARD, Computational Aerodynamics, Aeroelasticity

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18, NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** SURYA SURAMPUDI
a. REPORT UNCLAS	**b. ABSTRACT** UNCLAS	**c. THIS PAGE** UNCLAS	UL	18	**19b. TELEPHONE NUMBER** *(Include area code)* +44 (0)1895 616021

Parallel Bifurcation Solver and Model Reduction for Prediction of Wing Rock Onset

K.J.Badcock and M.A.Woodgate
Department of Engineering, University of Liverpool, Liverpool, U.K.

April 19, 2007

Abstract

This is the final report for the European Office of Aerospace Research and Development contract FA8655-06-1-3033, which started in March, 2006. Reports from a previous contract (FA8655-03-1-3044) covered the literature review and time domain solver, the formulation and coding of the bifurcation solver, and results from various methods for predicting wing rock onset, including time marching, the bifurcation solver, inverse power method and reduced modelling through Proper Orthogonal Decomposition. The work in the current period has involved the development and demonstration of a parallel bifurcation solver and the implementation and demonstration of reduced order modelling through centre manifold theory. Ken Badcock presented a paper on the parallel method at the Aerospace Sciences Meeting in Reno, January, 2006. After this a meeting was held at the Navy Academy with Phil Beran and others to discuss the exploitation of the techniques in Uncertainty Studies.

In accordance with the terms of the contract, the authors certify that there were no subject inventions to declare during the performance of this grant.

1 Introduction

Rolling motion about the longitudinal axis of a delta wing has been computed using Computational Fluid Dynamics (CFD) by several researchers [1, 2, 3, 4]. Recently a comprehensive numerical study of wing rock was conducted by Saad [7] using a three degree-of-freedom flight mechanics model for a generic fighter aircraft configuration (forebody, 65^o leading edge sweep, and vertical fin). Roll, sideslip and vertical degrees of freedom were included. Including the sideslip degree of freedom was found to delay the onset of wing rock and reduce the wing rock amplitude. These problems provide a useful test case for computing flight mechanics instability since they feature constrained motions and nonlinear aerodynamics.

The computational cost of time domain CFD based predictions can be high. A way of reducing the cost of computing parametric searches for aeroelastic stability was proposed by Morton and Beran[8]. Their method uses dynamical systems theory to characterise the nature of the aeroelastic instability, with this additional information concentrating the use of the CFD. In this way the problem of locating a one parameter Hopf bifurcation was reduced from multiple time marching calculations to a single steady state calculation of a modified (or augmented) system. This augmented system, solved via Newton's method, calculates the value of the parameter for which an eigenvalue of the system Jacobian matrix crosses the imaginary axis.

This approach was applied to compute the wing rock onset angle [14]. The formulation used a sparse matrix solver to efficiently solve the linear system arising at each Newton step for the augmented system. A matrix free product was used to evaluate the Jacobian-vector product for the augmented residual, resulting in convergence to the correct wing rock onset angle for a spatial discretisation of second order accuracy. However, the Newton iterations were driven to convergence by the Jacobian matrix of the first order scheme, leading to a loss of quadratic convergence. Note that the Jacobian matrix we refer to is of the coupled CFD-rolling motion system of equations, and so involves the derivatives of the CFD spatial discretisation (here based on Osher's approximate Riemann solver).

The classical shifted inverse power method (IPM) can be applied to the Jacobian matrix at any angle of attack to compute the interesting part of the eigenspectrum. This is useful to derive a starting solution for the augmented solver, and is a fast method for finding the onset angle in its own right. However, because only the Jacobian of the first order spatial scheme was available in the previous work it was not possible to exploit the full capability of the IPM.

Recently the analytical Jacobian for the second order spatial discretisation was derived and coded [15]. This opened up the possibility of applying the Inverse Power Method to calculate stability. This was exploited for predicting the wing rock onset angle.

This report describes the final approach taken to a parallel IPM solver, focussing on the details of the sparse linear solver. Results are presented to show the performance of the method for computing the wing rock onset angle. Then, the formulation and predictions of the reduced order model based on centre manifold theory are given.

2 Formulation

The coupled rolling motion/Euler equations can be written to emphasise the dependencies as

$$\frac{d}{dt}\begin{bmatrix} \mathbf{w}_a \\ \phi_t \\ \phi \end{bmatrix} = \begin{bmatrix} \mathbf{R}_a(\mathbf{w}_a, \phi, \phi_t, \alpha) \\ \mu C_{lc}(\mathbf{w}_a) + D\phi_t \\ \phi_t \end{bmatrix}.$$

Here the vector $\mathbf{R}_a$ denotes the discretisation of the spatial terms in the Euler equations, the details of which follow the formulation described in reference [18], which can be summarised as discretisation using Osher's method with MUSCL interpolation on moving multiblock meshes. The stability of an equilibrium $\mathbf{w} = \mathbf{w}_0$ which satisfies $\mathbf{R}(\mathbf{w}_0) = \mathbf{0}$ is determined by the eigensystem of the Jacobian matrix $A = \partial\mathbf{R}/\partial\mathbf{w}$. The matrix can be written out in block format as

$$A = \begin{bmatrix} \frac{\partial \mathbf{R}_a}{\partial \mathbf{w}_a} & \frac{\partial \mathbf{R}_a}{\partial \phi_t} & \frac{\partial \mathbf{R}_a}{\partial \phi} \\ \mu\frac{\partial C_{lc}}{\partial \mathbf{w}_a} & D & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} A_{aa} & A_{af} \\ A_{fa} & A_{ff} \end{bmatrix}. \tag{1}$$

The term A_{aa} is a large sparse matrix which represents the Jacobian of the discretisation of the fluid equations with respect to the fluid variables.

We consider the stability problem when the angle of attack α is the parameter. We are interested in finding the onset angle for the wing rock and assume that stability is lost through a Hopf Bifurcation. In this case the Jacobian matrix A has a pair of purely imaginary eigenvalues at the critical angle.

The Power Method [19] is an algorithm for calculating the dominant eigenvalue/eigenvector pair of any given diagonalizable matrix A. Its extension to the shifted inverse power method is practical for finding any eigenvalue provided that a good initial approximation is known. Assume that the $n \times n$ matrix A has distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ and consider the eigenvalue λ_j.

Then a constant ω can be chosen so that $1/(\lambda_j - \omega)$ is the dominant eigenvalue of the matrix $(A - \omega I)^{-1}$.

```
For initial guess $x_0$ and constant $\omega$
For $k = 1, 2, \ldots$ Do:
    $z_k = (A - \alpha I)^{-1} x_{k-1}$
    $\mu_k = ||z_k||_\infty$                                  (2)
    $x_k = z_k / \mu_k$
EndDo
```

The shifted inverse power method can be used to calculate the location of the critical eigenvalue in the complex plane at a fixed angle α. By computing the location for multiple values of α, the angle at which the eigenvalue crosses the imaginary axis can be computed. Variations on this approach are possible and are formulated in detail in reference [14].

3 Linear Solver - Serial

A key part of the computational work in the IPM is the solution of a large sparse linear system.

Eisenstat, Elman and Schultz [17] developed a generalized Conjugate Gradient method that depends only on A rather than $A^T A$ and is called the Generalized Conjugate Residual (GCR) algorithm. Saad and Schultz developed the Generalized Minimal Residual (GMRES) algorithm which is mathematically equivalent to GCR but is less prone to breakdown for certain problems and requires less storage and arithmetic operations. However GCR remains the easier algorithm to implement, especially in parallel. The GCR algorithm is:

```
$r_0 = b - Ax_0$
$p_0 = r_0$
For $j = 0, 1, 2, \ldots,$ until convergence. Do:
    $\alpha_j = \frac{\langle r_j, Ap_j \rangle}{\langle Ap_j, Ap_j \rangle}$
    $x_{j+1} = x_j + \alpha_j p_j$                                    (3)
    $r_{j+1} = r_j - \alpha_j Ap_j$
    $\beta_{ij} = -\frac{\langle Ar_{j+1}, Ap_i \rangle}{\langle Ap_i, Ap_i \rangle}$, for $i = 0, 1, 2, \ldots, j$
    $p_{j+1} = r_{j+1} + \sum_{i=0}^{j} \beta_{ij} p_i$
Enddo
```

To calculate the β_{ij} the vector Ar_j and the previous Ap_j's are required. The number of matrix vector products per step can be reduced to one if Ap_{j+1} is calculated by

$$Ap_{j+1} = Ar_{j+1} + \sum_{i=0}^{j} \beta_{ij} Ap_i \tag{4}$$

This may not be beneficial if A is sparse and j is large. A restarted version called GCR(m) is defined so that when the iteration reaches step m all the p_j's and Ap_j's are thrown away.

For the Block Incomplete Lower Upper (BILU) factorization the matrix is partitioned into 5×5 matrix blocks associated with each cell in the mesh. The use of this blocking reduces the memory required to store the matrix in a sparse matrix format. For clarity matrix elements refer to these blocks from now on.

Consider a general sparse matrix A whose elements are a_{ij}, $i, j = 1, \ldots, n$. A general incomplete factorization computes a sparse lower triangular matrix L and a sparse upper triangular matrix U so that the residual matrix $R = LU - A$ satisfies certain contraints, such as having entries in a prescribed pattern. A common constraint consists of taking the zero pattern of the L U factors to be precisely the zero pattern of A. However the accuracy of the ILU(0) incomplete factorization may be insufficient to provide an adequate rate of convergence.

More accurate block Incomplete LU factorisations allowing extra terms to be filled into the factorisation are often more efficient as well as more robust. Consider updating the a_{ij} element in full Gaussian Elimination (GE). The inner loop contains the equation

$$a_{ij} = a_{ij} - a_{ik}a_{kj}. \tag{5}$$

If lev$_{ij}$ is the current level of element a_{ij} then the new level is defined to be

$$\text{lev}_{ij} = \min(\text{lev}_{ij}, \text{lev}_{ik} + \text{lev}_{kj} + 1). \tag{6}$$

The initial level of fill-in for an element a_{ij} of a sparse matrix A is 0 if $a_{ij} \neq 0$ and ∞ otherwise. Each time the element is modified in the GE process its level of fill-in is updated by equation 6. Observe that the level of fill-in of an element will never increase during the elimination. Thus if $a_{ij} \neq 0$ in the original matrix A, then the element will have a level of fill-in equal to zero throughout the elimination process. The above gives a systematic algorithm for discarding elements. Hence ILU(k) contains all of the fill-in elements whose level of fill-in does not exceed k. The algorithm is given by:

```
For all non zero elements $a_{ij}$ define  lev $a_{ij} = 0$
For $i = 2, \ldots, n$ Do:
    For $j = 1, 2, \ldots, i-1$ and for  lev $a_{ij} < k$
        $a_{ij} = a_{ij}/a_{jj}$
        $a_{il} = a_{il} - a_{ij}a_{jl} \quad l = j+1, \ldots, n$          (7)
        Update the levels of fill in for non zero $a_{ij}$
    EndDo
    If  lev $a_{ij} > k$ then $a_{ij} = 0$
EndDo
```

There are two variations used for the preconditioning in the current work. First the linear solver described above is coded for complex variables. A complex variable formulation has 3 advantages over splitting the variable into real and imaginary parts. First it does not expand the bandwidth of the matrix. Secondly the precoditioner is faster to calculate and requires less storage. Thirdly the preconditioner was found to be more robust with respect to dropping off-diagonal terms.

The Block ILU(k) preconditioner of the Jacobian matrix is also available to precondition the system with the Jacobian matrix of the second order spatial scheme. The factorisation of the second order matrix can suffer from instability and often makes a poorer preconditioner than the factorisation of the matrix from the first order scheme. The first order matrix also has the advantage that it has 7 non-zero blocks per cell in the fluid grid, whereas the second order matrix has 13 non-zero blocks.

4 Linear Solver - Parallel

A key develop for making the IPM a practical proposition for full scale problems is the development of a parallel implementation of the method. Calculating the steady state and forming the Jacobian matrix are standard operations in parallel and will not be discussed. The difficult part is the linear solution and this is described in the current section.

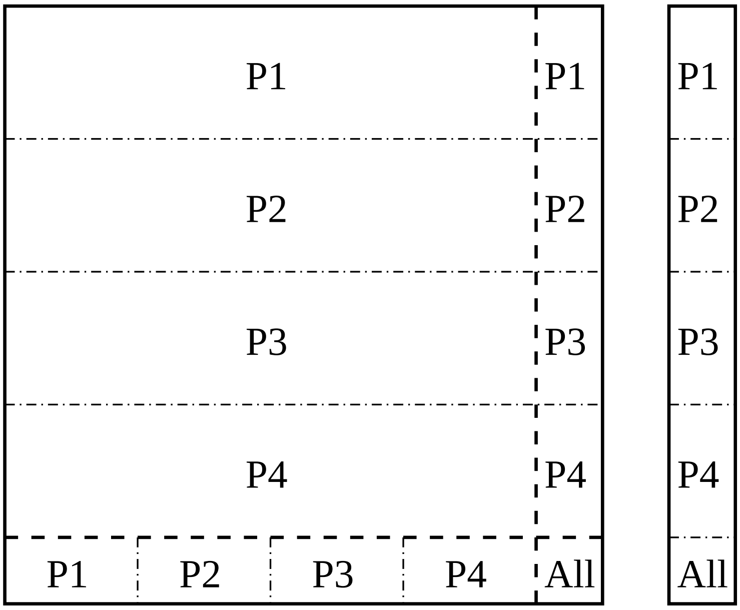


Figure 1: Effective data decomposition of the Jacobian and right hand side vector

4.1 Matrix Generation and Krylov Method

The data decomposition is done by storing whole blocks of the multiblock grid on a single processor. This means that the partitioning of the grid can be consided as a problem of partitioning the blocks. Each processor stores a certain number of blocks in the grid and their associated fluid variables. The structural variables are treated in a different way. Each processor stores all the structural information. This is currently not too expensive as the number of structural equations is small compared to the number of fluid equations, but might have to be revisited for different problems.

The residual and Jacobian are decomposed in the method shown in figure 1. Here 4 processors are assumed and the schematic in the figure indicates the system components

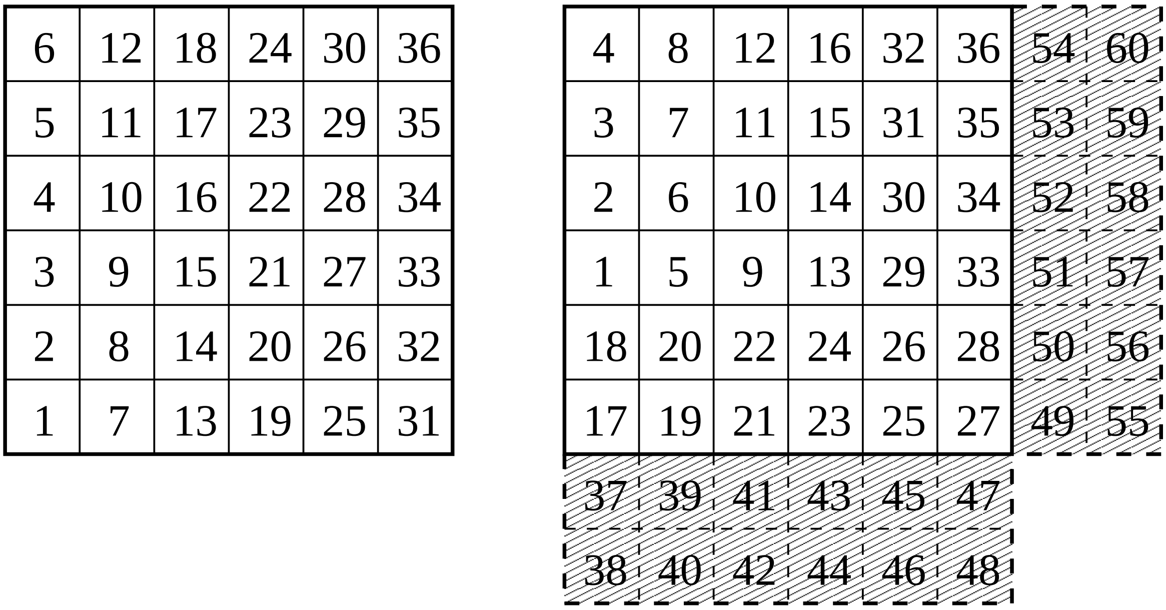
$$\begin{bmatrix} A_{ff} & A_{fs} \\ A_{sf} & A_{ss} \end{bmatrix} \begin{bmatrix} R_f \\ R_s \end{bmatrix}.$$

Processors with no solid boundary conditions will have zeros in the A_{sf} part of the Jacobian.

Since the Jacobian matrix is calculated in 4 distinct parts A_{ff}, A_{fs}, A_{sf} and A_{ss}, the matrix vector product is done in 4 parts which are summed together. The products with A_{fs} and A_{ss} are trivial as the structural unknowns are on all processors. The product with A_{sf} requires a “reduce all” global sum at the end of the multiply to get the information to all the processors.

The ordering of the fluid unknowns within each block is important. The cells are partitioned as shown in figure 2, with 3 different sets of cells. The matrix-vector product of the internal set can be updated without any communication and hence it is possible to complete this part of the product while waiting on the messages to arrive from other processors. The border set are the cells owned by this processor but which require some communication with other processors to be fully calculated. The external set is not updated at all on this processor but is used to update the border points.

To maximise the parallel performance of the code the blocks must be carefully assigned to the processors. If there is a large number of blocks it is not difficult to get approximately the same workload onto each processor for the CFD only case. However in the IPM the workload for non solid wall fluid points is greater than that for any other point - due to the contributions from



6	12	18	24	30	36
5	11	17	23	29	35
4	10	16	22	28	34
3	9	15	21	27	33
2	8	14	20	26	32
1	7	13	19	25	31

4	8	12	16	32	36	54	60
3	7	11	15	31	35	53	59
2	6	10	14	30	34	52	58
1	5	9	13	29	33	51	57
18	20	22	24	26	28	50	56
17	19	21	23	25	27	49	55
37	39	41	43	45	47		
38	40	42	44	46	48		

Figure 2: The renumbering scheme used in the Jacobian matrix A_{ff}

A_{sf}. Hence the number of fluid points on each processor and the number of solid wall points on each processor should be balanced. Lastly the number of processor block boundaries needs to be kept as small as possible. Not only does this keep the size of the messages small but much more importantly BILU(k) can be calculated with no extra appoximations on block boundaries if both blocks are on the same processor.

4.2 Preconditioning

Now, the crucial issue is the parallel implementation of the linear solver. The calculation of the LU factors and the forward and backward substitutions on these factors are both sequential operations. It is possible to ignore terms in the Jacobian matrix A which couple the system between processors. However, this can have a very bad influence on the convergence of the Krylov iterations for the systems being considered in the IPM. The decoupling of blocks through dropping terms localises the approximation to the solution of the linear system.

Calculating LU factors only has a limited amount of parallelism. In general a partitioning algorithm minimizes the number of interface elements by reducing the area of the boundaries. Then each processor can independently factor its set of nodes. This is a completely local operation. The interior nodes are then eliminated for the interface rows forming a reduced matrix A^I corresponding to interface nodes only. Finally all the processors factor A^I. This final operation is where all the difficulty in parallel sparse factorisations resides. The factorisation of A^I is done in stages in a nested fashion until A^I is factored. The full factorisation has not been implemented in the current work. The library EUCLID was used to test a parallel implementation of ILU and the performance of the algorithm was not found to be satisfactory due to the inherent limitations of the approach.

To improve the coupling between processors at the preconditioning stage a polynomial preconditioner was considered. In polynomial preconditioning the matrix M is defined by

$$M^{-1} = s(A)$$

where s is a low degree polynomial. The polynomial s is derived from the Neumann series expansion

$$I + N + N^2 + N^3 + \ldots + N^s \text{ where } N = I - A.$$

Then

$$M^{-1}A = (1 - N^{s+1}).$$

There is no limitation to implementing this preconditioner in parallel because it consists simply of matrix-vector products.

The option finally selected for the preconditioner used a combination of the localised BILU preconditioner with the first order polynomial preconditioner. The idea is that the polynomial preconditioner brings in some of the global information through matrix-vector products that is lost by the localisation of the BILU factorisation.

5 Test Case and Time Marching Results

There is a considerable amount of published experimental data for 80^o sweep delta wings. Hence this sweep angle was selected for the current study. The geometry of the wing was identical to that used by Arena and Nelson [5]. The wing has a flat upper and lower surface with a 45^o windward bevel and a root chord of 0.4222m. The moment of inertia for this wing was given as I_{xx} = 0.00125 Kg-m^2. The experiment was performed at a Reynolds number of 1.5×10^5. Since the flow solver here is compressible, a Mach number of 0.2 was used to avoid convergence difficulties at the very low Mach numbers used in the experiments. The freestream air density is used to non-dimensionalise the moment of inertia of the wing for the computations. The freestream air density in the experiment was unavailable and so the value was assumed to be $1.23Kgm^{-3}$ (sea-level ISA conditions).

A fine grid for the Euler simulations was created with approximately 1.6 million points. From this grid two levels were extracted by removing every second grid point in each direction. A RANS grid was also generated. The comparison of the pressure distributions along a spanwise cut with measurements from Arena and Nelson [5] is shown in figure 3. The RANS predictions agree well with the measurements, with only a slight under-prediction of the secondary separation. The medium grid Euler results predict the suction level close to the measured value. The Euler results do not predict the secondary separation, as this is associated with boundary layer separation. Simulations of free-to-roll motion at 15^o and 20^o angle of attack were conducted on the medium grid using the Euler equations. The roll angle histories for both simulations are shown in figure 4. At 15^o angle of attack the solution is dynamically stable (i.e. the amplitude of the oscillations decreases due to aerodynamic damping). However at 20^o angle of attack, the initial roll angle of 10^o initiates a wing rock response with the amplitude of the motion increasing with time.

6 Prediction of Wing Rock Onset Angle

6.1 Coarse Grid Tests

As a first test of the second order Jacobians and the linear solver, the IPM was used to calculate the wing rock onset angle on the coarse grid. 3 Levels of fill-in were used for the preconditioner which allowed the residual of the linear system to be driven down 11 orders in less than 60 iterations.

At 22 degrees, after 7 inverse power iterations the critical eigenvalue had converged to 6 significant figures. The value is $-1.381245 \times 10^{-4} \pm i0.223497$. At 23 degrees the converged value is $1.805895 \times 10^{-4} \pm i0.232095$. These angles bracket the onset angle, which can be estimated by linear interpolation as 22.4 degrees. This is in agreement with the results of other methods for the coarse grid [14].

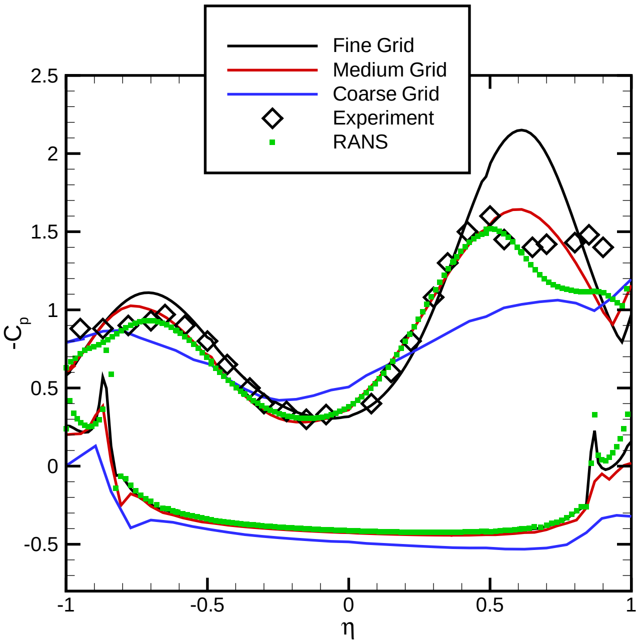


Figure 3: *Grid refinement study - Surface pressure distributions at 60%c_r. Lines indicate the coarse, medium and fine Euler results.*

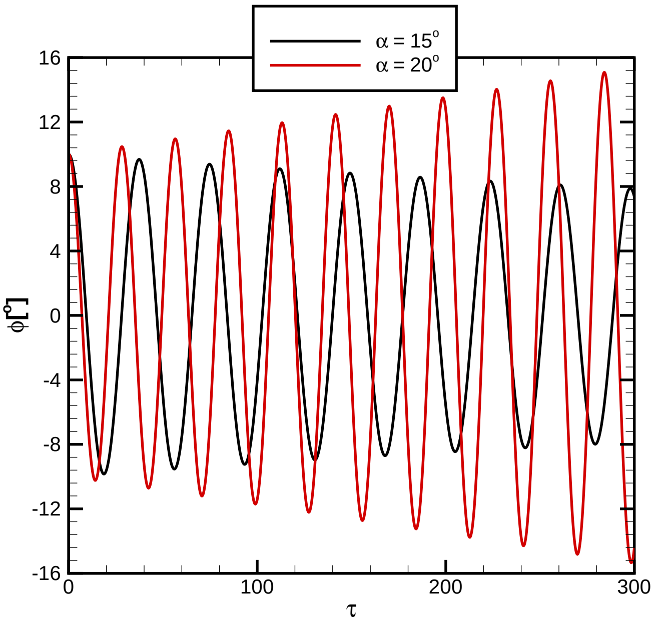


Figure 4: *Wing rock roll angle histories predicted using the Euler equations on the medium grid -* $\alpha = 15^o$ *and* 20^o

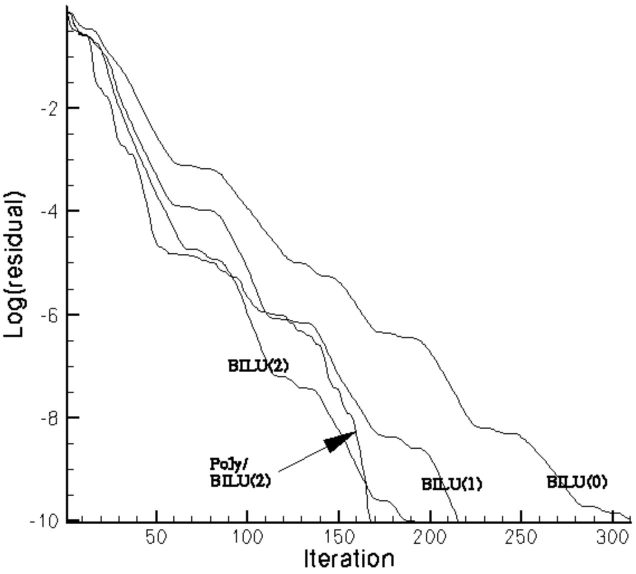


Figure 5: *Linear solver convergence for BILU(k) and Polynomial/BILU(2) on the coarse grid and 1 processors.*

Next, the use of the first order Jacobian to calculate the preconditioner for the linear system which uses the second order Jacobian is examined. The system used to test the linear solver was generated at 21 degrees incidence. There are two preconditioners used: the BILU factorisation with 1, 2 and 3 levels of fill-in, and the Polynomial preconditioner combined with BILU(2). The calculations shown in figure 5 were done on 1 processor and so the BILU factorisation involves no extra approximation. We first evaluate the results in terms of iterations to convergence since we are ultimately concerned with how the performance of the Krylov method scales in parallel. The number of iterations to convergence reduces as expected as the level of fill-in is increased. The number of iterations in each case is significantly larger than when the preconditioner is calculated for the second order Jacobian matrix, when 3 levels of fill-in are required. The increase in the number of iterations is from 60 to 168 although each iteration is around twice as expensive for the second order matrix preconditioner. Using the polynomial preconditioner on top of BILU(2) in this case does nothing to change the number of iterations to convergence.

The cost of computing the eigenvalues in terms of multiples of steady state calculation cost is 2.80 for BILU(0), 2.34 for BILU(1), 2.44 for BILU(2) and 3.70 for the Polynomial preconditioner in addition.

Next we test the scaling of the methods in parallel. Here the BILU factorisations are blocked on each processor and it is the influence of this that is the key concern. The convergence using BILU(2) with and without polynomial preconditioning on one and four processors is shown in figure 6. The number of iterations using the BILU preconditioner roughly doubles between 1 and 4 processors. Using the polynomial preconditioner actually reduces the number of iterations to convergence when moving from 1 to 4 processors. These results are encouraging in the sense that the polynomial precondioner is bringing in enough global information about the solution of the system to allow scaling to larger numbers of processors.

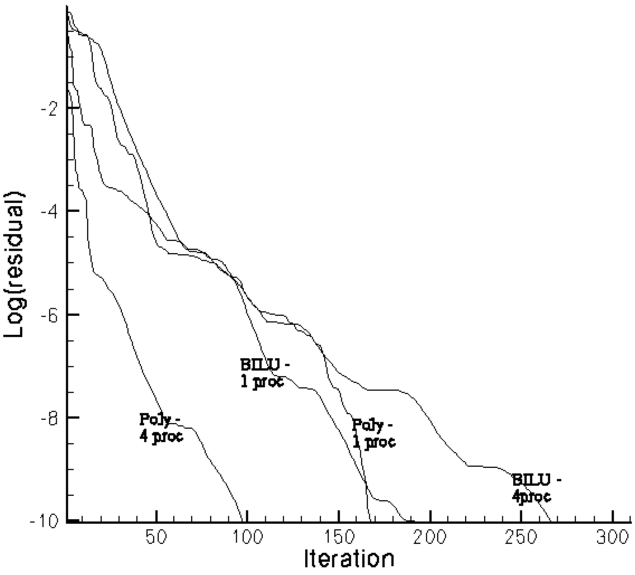


Figure 6: *Linear solver convergence for BILU(2) and Polynomial/BILU(1) on the medium grid and 1 and 4 processors.*

6.2 Medium Grid Tests

The test of the parallel solver is whether the medium grid can be solved, which it cannot in serial because of the memory requirements of the method (estimated 5.6Gb to store the Jacobian and preconditioner). To evaluate this the medium grid was calculated on 16 processors. The convergence of the linear solver is shown in figure 7. The system used here was generated at 15 degrees incidence.

First the BILU(2) results are shown for reference and the performance of the linear solver becomes very poor due to the localisation of the precondioner on each processor. The polynomial preconditioner significantly cuts the number of iterations to convergence. Results are shown using different levels of fill-in, and the number of iterations to convergence cluster between 200 and 260 iterations. The number of iterations will inevitably increase as the size of the grid increases, even sequentially.

The performance of the IPM is summarised in table 1. First, the improvement in the performance of the linear solver from using the polynomial preconditioner is clear. The number of linear solver steps to convergence is reduced by roughly a factor of 3. The resulting improvement in the time to convergence of the IPM is roughly 25-40%. Finally, the table shows the bifurcation happens between 16 and 17 degrees. This is consistent with the behaviour of the time domain solver shown in figure 4.

7 Reduced Order Model Implementation

7.1 Summary of Method

The wing rock onset angle is only one part of the information needed about the system. In addition the damping below the critical angle and the limit cycle amplitude above it are required. From the stability calculation we have the critical angle, the frequency and the critical eigenvector. Two

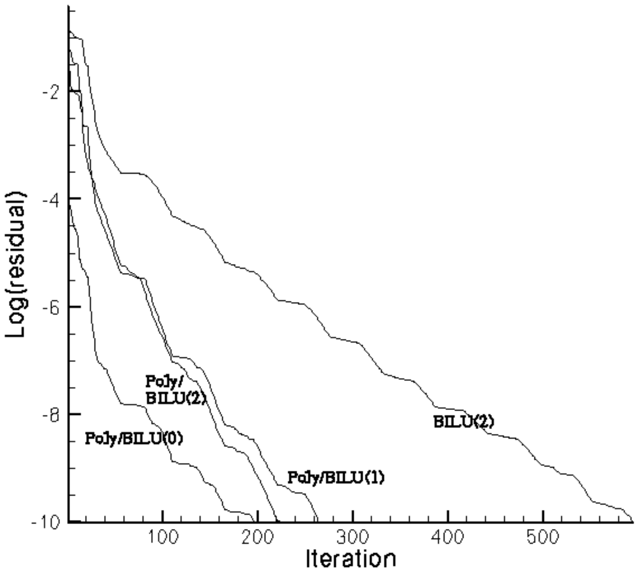


Figure 7: *Linear solver convergence for BILU(2) and Polynomial/BILU(k) on the medium grid and 16 processors.*

α	CPU Polynomial	CPU BILU(2)	Linear Polynomial	Linear BILU(2)	Eigenvalue
15^o	2.91	3.80	229	596	$\pm 0.172i - 7.53 \times 10^{-4}$
16^o	2.53	4.08	214	641	$\pm 0.183i - 2.25 \times 10^{-4}$
17^o	2.31	4.00	202	704	$\pm 0.192i + 2.01 \times 10^{-4}$
18^o	2.69	4.38	265	774	$\pm 0.202i + 6.32 \times 10^{-4}$

Table 1: Summary of Timings on Medium Grid on 16 processors

methods were presented in the previous report to exploit this information to develop a small order model for the prediction of this extra information.

The implementation of the first method is described below. This method has the following features:

- change of variables by projecting the full order solution onto the critical eigenvector
- Taylor expansion of the full order residual, up to and including third order terms
- Use of centre manifold theory to include the influence of the non-critical part of the solution on the critical part.

The current report describes the calculation of the difficult terms arising from the higher order derivatives, and then presents the initial results for the roll response.

7.2 Calculation of High Order Derivatives

As described in the previous report, the method of model reduction is

1. calculate α_0, $\mathbf{p}$, $\mathbf{q}$ and ω from the direct solver. Here $\mathbf{q}$ is the critical right eigenvector of the Jacobian matrix, $\mathbf{p}$ is the critical right eigenvector of the transpose of the Jacobian matrix, α_0 is the wing rock onset angle and $\pm i\omega$ is the critical eigenvalue.
2. calculate $B(\mathbf{q},\overline{\mathbf{q}})$ and $B(\mathbf{q},\mathbf{q})$. Here B is the second Jacobian operator
$$B(\mathbf{w}_1,\mathbf{w}_2)=\frac{1}{2}\frac{\partial^2 R}{\partial w^2}\mathbf{w}_1\mathbf{w}_2$$
3. solve
$$\left\{\begin{array}{rcl}(2i\omega I-A)h_{20} & = & B(\mathbf{q},\mathbf{q})\\ Ah_{11} & = & B(\mathbf{q},\overline{\mathbf{q}})\end{array}\right. \tag{8}$$
4. form $C(\mathbf{q},\mathbf{q},\overline{\mathbf{q}})$. Here
$$C(\mathbf{w}_1,\mathbf{w}_2,\mathbf{w}_3)=\frac{1}{6}\frac{\partial^3 R}{\partial w^3}\mathbf{w}_1\mathbf{w}_2\mathbf{w}_3$$
is the third Jacobian operator.
5. form g_{20}, g_{11}, g_{21} where
$$g_{20}=\langle\mathbf{p},B(\mathbf{q},\mathbf{q})\rangle \quad g_{11}=\langle\mathbf{p},B(\mathbf{q},\bar{\mathbf{q}})\rangle$$
and
$$\begin{array}{rcl}g_{21} & = & \langle\mathbf{p},C(\mathbf{q},\mathbf{q},\bar{\mathbf{q}})\rangle\\ & - & 2\langle\mathbf{p},B(\mathbf{q},A^{-1}B(\mathbf{q},\bar{\mathbf{q}}))\rangle+\langle\mathbf{p},B(\bar{\mathbf{q}},(2i\omega I-A)^{-1}B(\mathbf{q},\mathbf{q}))\rangle\\ & + & \frac{1}{i\omega}\langle\mathbf{p},B(\mathbf{q},\mathbf{q})\rangle\langle\mathbf{p},B(\mathbf{q},\bar{\mathbf{q}})\rangle\\ & - & \frac{2}{i\omega}|\langle\mathbf{p},B(\mathbf{q},\bar{\mathbf{q}})\rangle|^2-\frac{1}{3i\omega}|\langle\mathbf{p},B(\bar{\mathbf{q}},\bar{\mathbf{q}})\rangle|^2\end{array}$$
6. calculate the terms from the parameter expansion
$$\alpha_{00}=\langle\mathbf{q},\frac{\partial\mathbf{R}}{\partial\alpha}\rangle$$
$$\alpha_{01}=\langle\mathbf{q},A_\alpha\mathbf{p}\rangle$$
$$\alpha_{10}=\langle\mathbf{q},A_\alpha\bar{\mathbf{p}}\rangle$$
Note that the terms arising from
$$\alpha_2=\langle\mathbf{q},A_\alpha\mathbf{y}\rangle$$
lead to quadratic terms in z and $\bar{z}$ after exploiting of the centre manifold expression. These terms were found to be small and so were approximated as zero in the final calculations.
7. time march
$$\dot{z}=i\omega z+\frac{1}{2}g_{20}z^2+g_{11}z\bar{z}+\frac{1}{2}g_{02}\bar{z}^2+\frac{1}{2}g_{21}z^2\bar{z}+\hat{\alpha}(\alpha_{00}+\alpha_{01}z+\alpha_{10}\bar{z})$$

For the implementation a number of issues were tackled. First, the parameter terms, including the term A_α was calculated. This was done using finite differences, requiring only 2 residual evaluations.

Secondly, the second and third order Jacobian products were calculated using matrix free products against $\mathbf{q}$, $\mathbf{p}$ and their complex conjugates. This was tackled in previous work on aeroelastic instabilities. The first step is to rewrite the required derivatives in real and imaginary parts as follows Denoting
$$\mathbf{Q}=\mathbf{Q}_1+i\mathbf{Q}_2,\quad \mathbf{Q}\in C^n \quad \mathbf{Q}_1,\mathbf{Q}_2\in R^n.$$

then the following identities can be derived

$$\begin{array}{lcl} B(\mathbf{Q},\mathbf{Q}) & = & B(\mathbf{Q}_1,\mathbf{Q}_1) - B(\mathbf{Q}_2,\mathbf{Q}_2) + 2iB(\mathbf{Q}_1,\mathbf{Q}_2) \\ B(\mathbf{Q},\bar{\mathbf{Q}}) & = & B(\mathbf{Q}_1,\mathbf{Q}_1) + B(\mathbf{Q}_2,\mathbf{Q}_2) \\ C(\mathbf{Q},\mathbf{Q},\bar{\mathbf{Q}}) & = & C(\mathbf{Q}_1,\mathbf{Q}_1,\mathbf{Q}_1) + C(\mathbf{Q}_1,\mathbf{Q}_2,\mathbf{Q}_2) + iC(\mathbf{Q}_1,\mathbf{Q}_1,\mathbf{Q}_2) + iC(\mathbf{Q}_2,\mathbf{Q}_2,\mathbf{Q}_2). \end{array}$$

Further, the following identities can be derived:

$$\begin{array}{lcl} B(v+w,v+w) & = & B(v,v) + 2B(v,w) + B(w,w) \\ B(v-w,v-w) & = & B(v,v) - 2B(v,w) + B(w,w) \end{array}$$

so that $B(v,w)$ can be expressed as

$$B(v,w) = \frac{1}{4}\left[B(v+w,v+w) - B(v-w,v-w)\right].$$

A similar set of identities hold for C

$$\begin{array}{lcl} C(v+w,v+w,v+w) & = & C(v,v,v) + 3C(v,v,w) + 3C(v,w,w) + C(w,w,w) \\ C(v-w,v-w,v-w) & = & C(v,v,v) - 3C(v,v,w) + 3C(v,w,w) + C(w,w,w) \end{array}$$

and hence $C(v,v,w)$ can be expressed as

$$C(v,v,w) = \frac{1}{6}\left[C(v+w,v+w,v+w) - C(v-w,v-w,v-w) - 2C(w,w,w)\right].$$

By the use of directional derivatives it is then possible to evaluate the bilinear and trilinear functions $B(x,y)$ and $C(x,y,z)$ on any set of coinciding real vectors. These derivatives can be approximated using finite differences,

$$B(v,v) = \frac{1}{h^2}\left[\mathbf{R}(\mathbf{w}_0 + hv, \mu_0) - \mathbf{R}(\mathbf{w}_0 - hv, \mu_0)\right] + O(h^3) \tag{9}$$

and

$$C(v,v,v) = \frac{1}{8h^3}\left[-\mathbf{R}_3 + 8\mathbf{R}_2 - 13\mathbf{R}_1 + 13\mathbf{R}_{-1} - 8\mathbf{R}_{-2} + \mathbf{R}_{-3}\right] + O(h^4) \tag{10}$$

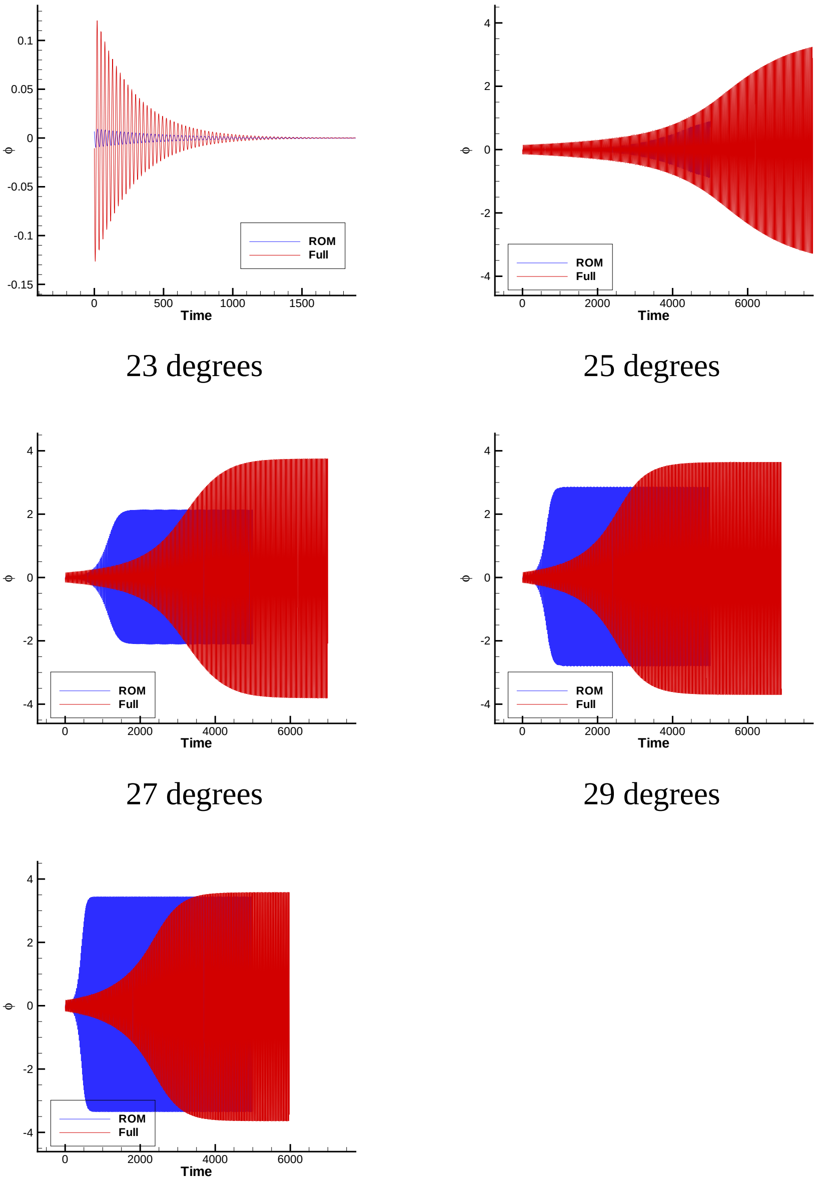
where h is small and $\mathbf{R}_l = \mathbf{R}(\mathbf{w}_0 + lhv, \mu_0)$.

The evaluation of these finite differences suffers from truncation error for values of h which are too large, and from rounding error for values which are too small. To overcome the second limit, quad-double arithmetic was used for the residual evaluation. A high precision version of algebraic and transcendental functions is also be required, in this case because of the contributions of such functions in Osher's flux function. The QD library [23] was used to obtain this functionality. This library allows extension of existing code to double-double precision and quad-double precision without major recoding, by using operator overloading.

To test the accuracy of the reduced model terms, the convergence of the third Jacobian term, for the second order spatial discretisation, with h is considered. The third Jacobian term is more sensitive to rounding error than the second term, and also the parameter terms. The convergence is shown in table 2 and there is a good range of values for h where satifactory values are obtained.

7.3 Limit Cycle Predictions

The model reduction was exercised on the coarse grid using the first order spatial scheme. The onset angle was previously computed as 24.3 degrees. The model coefficients were computed and the two degree of freedom model was run for different values of $\bar{\alpha} = \alpha - 24.3$. The comparison with the full order model is shown in figure 8.



23 degrees

25 degrees

27 degrees

29 degrees

31 degrees

Figure 8: Comparison of the response of the full order and reduced order models on the coarse grid using a first order spatial discretisation.

h	Re $\langle \mathbf{q}, C(\mathbf{p}, \mathbf{p}, \bar{\mathbf{p}}) \rangle$	Im $\langle \mathbf{q}, C(\mathbf{p}, \mathbf{p}, \bar{\mathbf{p}}) \rangle$
4	5.5347700126789234e-01	6.2858220561002369e+00
6	3.5505862837191756e+02	2.3350599906148987e+03
8	3.1526220567128992e+04	1.6834102453444750e+05
10	1.9147851302829609e-04	1.1694372975048235e-02
12	1.9147851302829441e-04	1.1694372975048255e-02
14	1.9147851302817110e-04	1.1694372975048051e-02
16	1.9147858912814422e-04	1.1694373075537749e-02

Table 2: Convergence of reduced order model third Jacobian contribution with finite difference step size. These results are on the coarse grid using the second order spatial discretisation.

Finally, the reduced model predictions for the second order spatial scheme on the coarse grid were calculated. The growth of the limit cycle oscillation compared with time domain results is shown in figure 7.3. For the full order predictions the calculations took 1-2 days to settle to a limit cycle oscillation. The reduced model predictions were obtained almost instantaneously for any angle after the reduced model ccoefficient were derived, an operation that takes no more than 30 minutes using a laptop for this grid. The comparison is excellent within a few degrees of the wing rock onset angle and gradually becomes worse as the angle of attack becomes larger. It is expected that the Taylor series and centre manifold approximations will become worse as we move away from the bifurcation point. However, the most useful information is close to the onset angle, and the reduced model tells use the expected amplitude immediately after the onset angle is passed.

8 Conclusions

Two achievements have been reported. First, a parallel version of an eigenvalue based stability prediction method has been demonstrated. The key development was to use a polynomial based preconditioner on top of a localised BILU factorisation to reduce the effect of the localisation on the convergence of the linear solver. Calculations were successfully carried out on the medium delta wing grid on 16 processors. The performance of these calculations showed that the eigenvalue at each angle of incidence could be computed in a computational cost between 2 and 3 times the cost of a steady state CFD calculation. It seems likely that better forms of the polynomial preconditioner could be developed that take advantage of the fact that we know the eigenvalue that the IPM is converging to is also an eigenvalue of the coefficient matrix if the linear system. It seems likely that this slows down the convergence of the linear solver.

Secondly, the application of the projected reduced order model to wing rock limit cycle prediction was made. Key difficulties for the wing rock problem was the need to use the angle of attack as the bifurcation parameter. This means that the parameters terms in the reduced model needed to include derivatives of the CFD residual function and its Jacobian with the respect to the angle of attack. The predictions from the two degree of freedom ROM of the limit cycle oscillation amplitudes were in excellent agreement with calculations using the full order system, at dramatically reduced computational cost.

Future work would involve applying these techniques to full sized aircraft problems, and to integrating the methods into a framework for parametric searches for stability (both aeroelastic and flight mechanics), including the assessment of uncertainty.

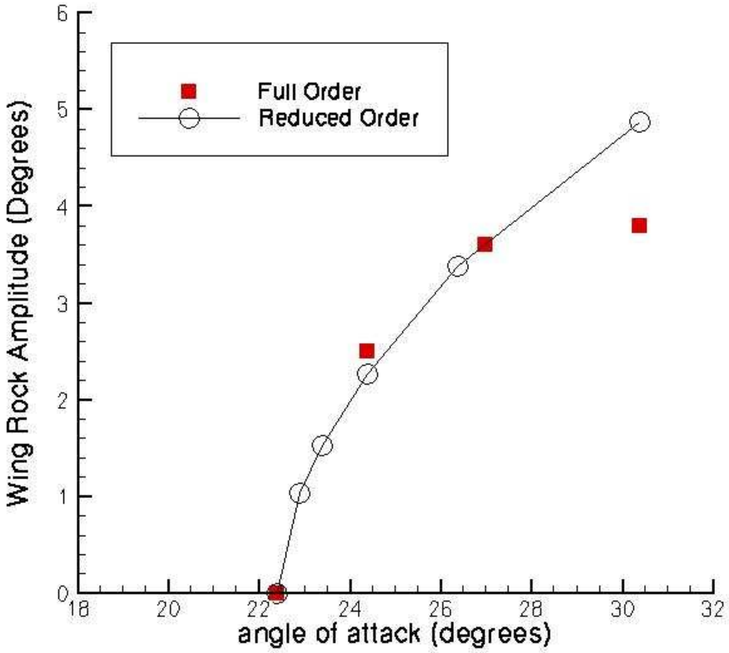


Figure 9: Comparison of the predicted Wing Rock amplitudes of the full order and reduced order models on the coarse grid using a first order spatial discretisation.

9 Acknowledgements

This work has been funded by the European Office of Aerospace Research and Development under contract FA8655-03-1-3044.

References

[1] Lee-Rausch, E. M., and Batina, J. T., “Conical Euler analysis and active roll suppression for unsteady vortical flows about rolling delta wings”, NASA Techical Paper 3259, 1993.

[2] Chaderjian, N. M., and Schiff, L. B., “Numerical simulation of forced and free-to-roll delta wing motions”, J. Aircraft, 33(1), pp. 93-99, Jan. - Feb. 1996.

[3] Fritz, W., “WEAG TA 15 Common Exercise V - Numerical and experimental investigation of a delta wing in rolling motion”, DASA Report, DASA-S-R-1820, April 2000.

[4] Soemarwoto, B., Boelens, O., Fritz, W., Allan, M., Ceresola, N., and Buetefisch, K., “Towards the simulation of unsteady maneuvers dominated by vortical flow”, AIAA-2003-3528, June 2003.

[5] Arena, A. S. and Nelson, R. C., “Experimental investigations on limit cycle wing rock of slender wings”, J. Aircraft, 31(5), pp. 1148-1155, Sept. - Oct. 1994.

[6] Levin, D., and Katz, J., “Self-induced roll oscillations of low-aspect-ratio rectangular wings”, J. Aircraft, 29(4), pp. 698-702, July-Aug. 1992.

[7] Saad, A. A., “Simulation and analysis of wing rock physics for a generic fighter model with three degrees-of-freedom”, PhD Dissertation, Air Force Institute of technology, Wright-Patterson Air Force Base, Dayton, OH, July 2000.

[8] Morton, S.A. and Beran, P.S., Hopf-Bifurcation Analysis of Airfoil Flutter at Transonic Speeds, J Aircraft, 36, pp 421-429, 1999.

[9] Beran, P.S. and Carlson, C.D. Domain-Decomposition Methods for Bifurcation Analysis, AIAA Paper 97-0518, 1997.

[10] Beran, P.S., A Domain-Decomposition Method for Airfoil Flutter Analysis, AIAA Paper 98-0098, 1998.

[11] Badcock, K.J., M.A. Woodgate, M.A. and Richards, B.E., The Application of Sparse Matrix Techniques for the CFD based Aeroelastic Bifurcation Analysis of a Symmetric Aerofoil, AIAA J, 42(5), 883-892, May, 2004.

[12] Badcock, K.J., Richards, B.E., and Woodgate, M.A., “Elements of Computational Fluid Dynamics on block structured grids usin g implicit solvers”, Prog. Aerospace Sci., Vol. 36, pp 351-392, 2000.

[13] Badcock, K.J., Woodgate, M.A. and Richards, B.E., Direct Aeroelastic Bifurcation Analysis of a Symmetric Wing Based on the Euler Equations, Journal of Aircraft, 42(3),731-737, 2005.

[14] Badcock, K.J., Woodgate, M.A. and Beran, P.S., CFD Based Wing Rock Stability Analysis using Time Marching, Reduced Modelling and Hopf Bifurcation Methods, International Forum on Aeroelasticity and Structural Dynamics, Munich, July, 2005.

[15] Woodgate, M.A. and Badcock, K.J., On the fast prediction of Transonic Aeroelastic Stability and Limit Cycles, in review for AIAA Journal, 2006.

[16] Osher,S. and Chakravarthy, S.R., Upwind Schemes and Boundary Conditions with Applications to Euler equations in General Coordinates , Journal Computational Physics, vol. 50 , 1983, p 447-481.

[17] Eisenstat, S.C., Elman, H.C. and Schultz, M., Variational Iterative Methods for Nonsymmetric Systems of Linear Equations, SIAM Journal of Numerical Analysis, Vol. 20, No. 2, April 1983, 345-357.

[18] Badcock, K.J., Richards, B.E. and Woodgate, M.A., Elements of Computational Fluid Dynamics on Block Structured Grids using Implicit Solvers, Progress in Aerospace Sciences, vol 36, 2000, pp 351-392.

[19] Golub, G.H. and Van Loan, C.F., Matrix Computations, The John Hopkins University Press, 1996.

[20] Tuminaro, R.S., Heroux, M., Hutchinson, S.A. and Shahid, J.N., Official Aztec User’s Guide Version 2.1, SAND99-8801J, Sandia Laboratory, 1999. Official Aztec User’s Guide Version 2.1, SAND99-8801J, Sandia Laboratory, 1999.

[21] Axelsson, O., Iterative Solution Methods, Cambridge University Press, 1994.

[22] Arena, A., “An experimental and computational investigation of slender wings undergoing wing rock”, PhD thesis, University of Notre Dame, IN, USA, April 1992.

[23] crd.lbl.gov/ dhbailey/mpdist/index.html